

# forse über ACD2

- 1 -

- Kovalter Algorithmen LCS mit D&C und Mem(x<sup>i</sup>, y<sup>j</sup>, mem)

else {

if (mem[i-1][j] = -∞) {  
 mem[i-1][j] = LCSmitD&CundMem(x<sup>i-1</sup>, y<sup>j</sup>, mem)}

if (mem[i][j-1] = -∞) {

mem[i][j-1] = LCSmitD&CundMem(x<sup>i</sup>, y<sup>j-1</sup>, mem)}

if (mem[i-1][j-1] = -∞) {

mem[i-1][j-1] = LCSmitD&CundMem(x<sup>i-1</sup>, y<sup>j-1</sup>, mem)}

mem[i][j] =

return max{mem[i-1][j],

mem[i][j-1],

mem[i-1][j-1] + u(x<sub>i</sub>, y<sub>j</sub>)}

}

...  
...

- Branch & Bound für Rucksack

Input:  $z_1, \dots, z_n, Z, p_1, \dots, p_n, P, l, x_j = b_j$  für  $j = 1, \dots, l-1$

Output: Opt. Lösung mit  $x_l = b_{l-1}, \dots, x_{l-1} = b_{l-1}$

① if  $(\sum_{j=1}^{l-1} b_j z_j > Z)$  then return

② if ( ~~$l > n$~~ ) then return

③ if  $(\sum_{j=1}^{l-1} b_j z_j > P)$  then

$$P := \sum_{j=1}^{l-1} b_j p_j$$

④ Compute  $U := UB(b_1, \dots, b_{l-1})$

④.5 Compute  $P_i := LB(b_1, \dots, b_{l-1})$  } greedy mit min  
ganz. lag. } max

⑤ if ( $U > P$ ) then {

$b_l := 1$  Branch & Bound (..., l+1, ...)  
 $b_l := 0$  ..., l+1, ...)

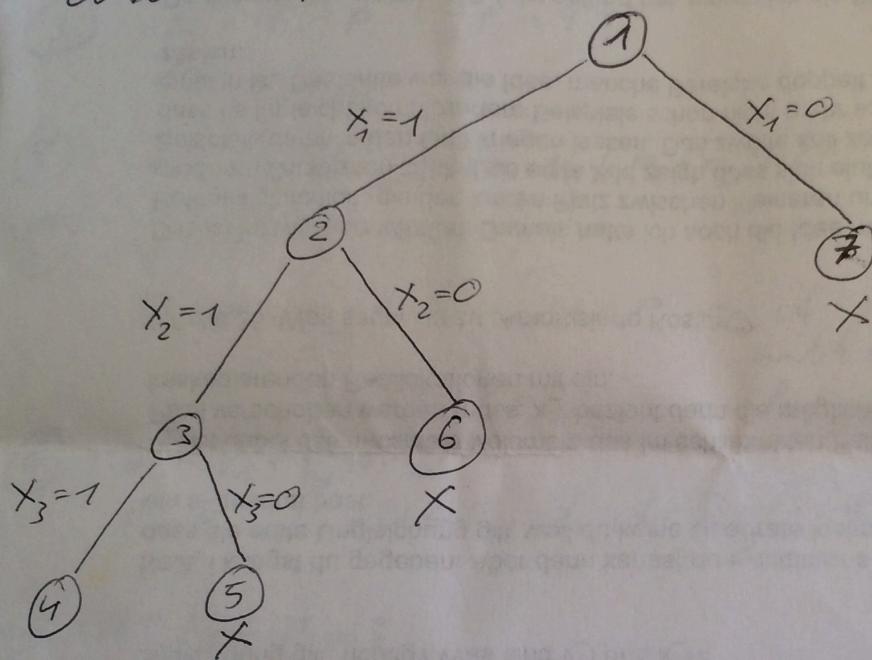
⑥ return

Bsp.:	i	1.	2	3	<del>4</del>
$z_i$	2	1	3	<del>4</del>	<del>4</del>
$p_i$	2	2	1	<del>4</del>	<del>4</del>
$\frac{z_i}{p_i}$	1	$\frac{1}{2}$	3	<del>4</del>	<del>4</del>
$\bar{u}(i)$	2	2	3	<del>4</del>	<del>4</del>

$$z = \cancel{4}$$

- 2 -

Entscheidungsbaum



Knoten:

(1) :  $U = P_2 + P_1 + \frac{1}{3}P_3 = 4\frac{1}{3}$   
 $P = P_2 + P_1 = 4$

(2) :  $U = P_2 + P_1 + \frac{1}{3}P_3 = 4\frac{1}{3}$   
 $P = P_2 + P_1 = 4$

(3) :  $U = P_1 + P_2 + \frac{1}{3}P_3 = 4\frac{1}{3}$   
 $P = P_2 + P_1 = 4$

(4) :  $U, P$  undefined, da  $P_1 + P_2 + P_3 > P$

(5) :  $U = P_1 + P_2 = 4$   
 $P = P_1 + P_2 = 4$        $U = P \Rightarrow$  if-Bd. in (5)  
nicht erfüllt  
 $\Rightarrow$  keine rek. Aufrufe

(6) :  $U = P_1 + P_3 = 3$   
 $P = 4$        $U < P \Rightarrow$  if-Bd. in (5) nicht erfüllt  
 $\Rightarrow$  keine rek. Aufrufe

(7) :  $U = P_2 + P_3 = 3$   
 $P = 4$       = if-Bd...  
 $\Rightarrow \dots$